Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Thinking about tasks

To build a successful interface and write successful documentation, you need not only to know about your users. You also
need to know about your users' work. What do they do? What goals are they trying to accomplish? What tasks do they
now do to meet those goals? What tasks will your product help them do?
How do they actually perform the tasks today? What problems do they have performing those tasks? Where can you see
ways to simplify what users do so that they can accomplish their goals more easily? How do the tasks that one person
does relate to tasks that others do to accomplish a, given piece of work?
How do users differ in terms of the tasks they do or the way they use a product because of how long they've been using it,
how often they use it, how comfortable they are with it-that is, what's the difference between novices and experts, and
what other "stages of use" are there between those two?
To answer these questions, you need to do task analysis-and that's what we discuss in this chapter. We begin with the
question "What is task analysis?" and go on to consider three main topics:

- starting with users' goals
- identifying different types and levels of task analysis
- thinking of users according to their stages of use

We describe the many types of task analysis that you should consider, including

- workflow analysis
- job analysis
- combining workflow analysis and job analysis
- task lists and task inventories
- process analysis, task sequences
- task hierarchies
- procedural analysis

What is task analysis?

If we are talking about goals and tasks, we are talking about task analysis. The point of any and all parts of a
product-software, hardware, interface, documentation-is to help people do things. Usually we can think of those "things"
as work. The work may be to

- admit a patient to the hospital
- find a customer's order in a database
- send a message to everyone on a project team
- put up a new Web site
- change payroll codes for an employee who has moved
- set up a new computer at home

Sometimes those "things" people do with what you are developing aren't "work" in the traditional sense of doing
something that brings in money. The things people do with your product may be for personal benefit, leisure activities, or
social interactions. For example, users may want to

- browse the Internet, looking for interesting sites
- take pictures of the children to make the grandparents happy
- cook a fancy dinner to impress a friend
- listen to a new recording while sipping wine with someone special
- record a program on the VCR to watch at a later time

Although these are leisure tasks, the users who are trying to do them still have goals (have fun, make others happy,
impress that special someone) and are still doing tasks to meet those goals. Task analysis is as applicable to these
leisure activities as it is to work activities.

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Taking a broad view of task analysis

In this book we take a very broad view of task analysis. In addition to applying task analysis to both work and leisure, we
also consider as types of task analysis all levels of generality and detail from people's goals to the lists of tasks they do to
the specifics of the actions they take and the decisions they make. You may have heard the term "task analysis" used
primarily to describe the most detailed level in which a task is decomposed into steps and decisions. We call this detailed
level "procedural analysis" in this book.

However, we argue in this chapter that to build a successful product you must start by considering users' goals. In new
designs, the list of tasks users do may change. The procedures (specific steps) they take to do those tasks may change.

But users' goals are much less likely to change. If you don't understand the users' goals, you may well design a product
with simple procedures that users have no interest in using. Task analysis includes understanding users' goals.

We also argue that, at the predesign stage, many types of task analysis are relevant and perhaps more relevant than the
traditional view of task analysis as elaborating the procedural details. That's why we discuss a variety of types and levels
of task analysis, including looking at workflow analysis covering the work of more than one user, looking at a single user's
entire job, gathering task lists, and seeing processes at different levels of detail.

Understanding task analysis at the predesign stage

If you have come to this book with a background in documentation or training, you may already know about task analysis
as an important step in understanding what to put in a manual or in training materials. But there is a critical and
fundamental difference between the task analysis that we are talking about in this book and the task analysis that you
might do later to know how to explain a procedure or train someone to work with a product.

Here the goal is to figure out how to design or redesign a product. The new product doesn't exist yet. You don't know what
the procedures are. You are working with users and collecting data in order to decide what tasks the product should
support and what procedures should be built into the product.

That means that you want to observe real users in all their messy reality. You want to observe, listen to, and talk with
users at all stages of use, including novices and advanced beginners, not only competent users and experts. You want to
see how these real users now do what they do, even if they don't do it efficiently, even if they make mistakes. In fact, you
want to see all the errors, workarounds, and problems that they have. That's all information that can help you help them
with your new design.

Documentation and training specialists have an important role in predesign task analysis, in part because they know
about task analysis. They should be part of the team from the beginning because designing the interface must include
decisions about where and how the product will communicate with users (from all the words and images in the interface to
built-in performance and training aids to online help and tutorials to paper documents). Task analysis before design
informs decisions about which types of communication are needed for which users and how to organize each type of
communication.
However, many books on task analysis cover only the types of task analysis that documentation and training specialists
typically do after the product is designed. At that point the new procedures are in place, and the goal of task analysis is to
understand how to help someone who does not know the new procedures learn to do them. For that type of task analysis,
you usually observe someone who knows the procedures and who doesn't make mistakes doing them. You are usually
interested in the low-level details of each step that users take. You usually select the more expert performers to observe
for your task analysis.
Task analysis for use in designing interfaces doesn't always lend itself to the straightforward flowcharts and neat lists that
you can get after the product is designed. It's much messier, but it's a critical stage in successful product design.

Starting with users' goals

To do a task analysis, you should understand users' goals and how users move from goals to tasks to actions. For
example, users inside a company may have goals like these:

- keeping my job
- getting done so I can go home on time
- making the boss happy so I get a good performance review

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Users don't often talk about goals at this level. They usually focus on more specific work goals like "set up my calendar to
be an effective planning tool." Nonetheless, users have goals and values like these. Products succeed when they help
users meet their goals, and they don't succeed when they make the goals more difficult or impossible.
Companies also have goals for users doing tasks. Companies may have goals like these:

- increasing revenue
- increasing the number of applications that get processed in a day
- decreasing the cost of providing support

Sometimes the company's goals and the users' goals match. In the case of the three goals we just listed, we can make
the argument that improving the usability of products will help the company meet those goals and also allow users to meet
their goals that relate to doing a good job so as to get good performance reviews, etc.
However, in our experience, all too often products are designed only to meet company goals, or decisions about buying
products from outside are made only in terms of company goals. The implementation makes it impossible for users to
meet both their goals and the company's goals. Take the example of a company that puts into place a new

computer-based, time tracking system in which employees are supposed to account for their work hours by charging each
hour of each day to specific codes depending on what they actually do during that time. The company's goals are to stay
out of trouble with the auditors, to have greater accountability, and to get better data to help with future strategic planning.
The employees' goals are to get paid regularly, to go home on time, and to complete the substantive project work that is
their primary responsibility.
If the computer-based system is difficult to use, employees can't meet both their goals and the company's goals. We've
heard of systems in which employees don't put in the data that upper management expects to get out of the new system.
It takes so much time to figure out how to do anything out of the ordinary that users assign all their work to the same code
week after week. If they spent more time figuring out the system, they would work overtime for which they don't get paid.
Time spent with the time tracking system is time not spent on doing their project work, but their performance ratings
depend on how much project work they get done. Employees get their paychecks as long as they do fill out the computer
worksheet every week, no matter what codes they put in. The employees are using the product in the ways that best meet
their goals, but they aren't doing what management wants them to do. Because the developers didn't take users' goals
into account in designing the new system, the company's goals aren't being met.
Successful products are designed by understanding both user goals and company goals (or parents' goals and children's
goals or buyers' goals and sellers' goals). Goals give you users' values. Respecting those values when you design is the
only way to make sure products succeed. As we look at users' goals, we discuss these topics:
• relating goals to tasks to actions
• seeing how users choose tasks to meet goals
• seeing what happens when users have problems
• keeping goals as part of task analysis

Relating goals to tasks to actions

A task is what someone does to achieve a goal. As Donald Norman explains (1988, 46), "to get something done, you
have to start with some notion of what is wanted-the goal that is to be achieved:' It is true, as he also says (1988, 49), that
we cannot always articulate our goals clearly, but when we do tasks we are working towards goals, such as
• finishing this chapter before the deadline
• getting dinner on the table before the hungry children complain
• sending some information to a colleague in a different country quickly
Norman also gives us a picture of how people go about accomplishing their goals. Figure 3-1 shows his seven-stage
action cycle, in which Norman takes us through the major steps that we all use in our interactions with all sorts of products
from documentation to software to machines like stoves and overhead projectors to other types of objects like doors and
lightbulbs. Every one of these products has an interface that people have to know how to use to do tasks and accomplish
goals. The goal might be finding an answer (from the document) or communicating with a friend (through electronic mail

software), making tea (by boiling water on the stove), providing information (by showing a viewgraph on the overhead
projector), getting into a room (through the door), or having enough light to read this book (by using the light bulb). All of
Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish
these products (document, electronic mail, stove, overhead projector, door, lightbulb) are either facilitators or obstacles for
users who are trying to do tasks so they can meet their goals.

1 Forming the goal
2 Forming the intention
3 Specifying an action
4 Executing the action
5 Perceiving the state of the world
6 Interpreting the state of the world
7 Evaluating the outcome

Figure 3-1 Norman's seven-stage cycle of how people behave in terms of achieving goals and performing tasks.

Let's consider an example of a simple situation and follow it through Norman's seven stages, as shown in figure 3-2.

1. User forms goal. Go outside to et some fresh air.
2. User forms intention decides task). Open the door.
3. User specifies actions . "It looks like I pull this handle here."
4. User does the actions . pulls on the handle.
5.User perceives the state of the world. The door didn't open.
6. User interprets the state of the world. "Well. That didn't work. This handle sure looks like I should pull it. I guess
it doesn't mean that. I ue55 I need to push it."
7. User evaluates the outcome.

Didn't get outside yet. (If the user still wants to meet the goal and still
thinks this task is the best way to do it, the user goes through steps 3-7
again this time pushing on the door.

Figure 3-2 Users go from goal to task to action to interpreting what happened.

Seeing how users choose tasks to meet goals

Norman's view of the action cycle is useful because it makes us realize that people often have many options for the tasks
they can do to achieve goals (see figure 3-3). For example, to get dinner on the table before the hungry children complain,
the person responsible for dinner might use the stove, the microwave, the telephone (to order carry-out), the car (to get
the carry-out or to stop and pick up something on the way home from work), combination of these options. What that
person chooses depends on how he or she weighs factors that are exiterna how the task is done but that are an i nt part
of a task analysis. These factors include time, cost, the person's skills and confidence in the different methods, the ease
for that person of learning and using a particular method, and the value that person-and the customers, in this case, the
hungry childrenplaces on these and other factors, such as, in this case, taste and nutrition.
Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Figure 3-3 Users often have options of what tasks to do to reach their goal.

Take the example that brings us closer to our interest in software and documentation: To get information to a colleague in
another country, the sender might be able to put it in electronic mail (e-mail) as an unformatted message, send it by e-mail

as a formatted attachment, fax it through the computer, print it and send it through a separate fax machine, mail it, send it by overnight service, or get on a plane and hand-deliver it.

When the sender weighs cost and time, some of these options are clearly more favorable than others. Getting on a plane and hand-delivering a message to another country isn't a very feasible option for most of us. However, most of us are likely to factor in the time it takes to learn the tools required (hardware, software, and documentation) to use some of these methods if we don't already know them. For example, even though the sender's e-mail program allows attachments, if this is the first time the need for an attachment has come up, that user may decide that it's too much trouble to figure out how to send one and instead may just put the document into the message as unformatted text. The sender might spend a few minutes trying, but without quick success may give up and settle for a less-desirable but easier solution to reach the goal.

If you are designing a new e-mail program and want people to use it to send formatted attachments, you have to make learning that function easy enough to meet the typical user's self-imposed constraint on how much time it is **worth to learn** how to use the attachment function. You learn those users' constraints by observing them on the job doing similar tasks, if your product doesn't yet exist.

Both of these cases show that to design a product that people will use, you have to know the users' goals. You also have to know what matters most to both the decision makers who buy your product and the users who actually use it. That is, you have to know what they value (low cost, reliability, ease of learning, speed, etc.). You have to know what trade-offs they will make in deciding when and how to do different tasks to accomplish their goals. You also have to know about the people and environments on both ends of a process. For example, in our second case of getting a message quickly to someone in another country, you would also need to know about the technology that is available to the person who is receiving the message. All that should be part of a task analysis.

Seeing what happens when users have problems

Users may also change the task, change the goal, or give up. If users have problems getting through the action cycle, they may change the way they try to achieve the goal. A user who cannot quickly figure out how to use a computer fax program may change the task and print out the document for the fax machine instead. If no fax machine is available, the user may decide that the colleague doesn't need the information *immediately,* changing the goal to "getting information to the colleague in a few days when I can find someone who knows how to use the computer's fax program." The user who gets really frustrated may give up and decide that the colleague doesn't need the information after all.

Users decide when and how to use documents, software, and hardware. Understanding users' tolerances for time and effort for different tasks and different technologies is part of task analysis. It helps you set usability goals for design, which are discussed in chapter 12.

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Keeping goals as part of task analysis

Some human-computer interaction (HCI) specialists suggest that task analysis must be *device dependent* (Benyon 1992). [See the reply by Diaper and Addison (1992)]. By "device," Benyon means the equipment, platform, tool, or product that will be used for the task. He suggests that a task analysis must be done within the context of a specific situation. The problem is one of definition, of which levels we accept as being part of a task analysis.

Benyon calls the goal level (Norman's first stage) the "external task," and he would exclude it from a task analysis.

He would start the task analysis with what he calls the "internal task;" which begins only after the user has formed the intention in Norman's terms.

However, if we start at Norman's second stage, we are missing a very important part of how users work. We are missing the essential connection between the users' goals and the specific ways of meeting those goals. If we start task analysis only with Norman's second stage, we run the risk of being device driven in the design rather than being user centered. Because users are goal oriented, the point of design must be to find the best device or program to help users achieve their goals.

In fact, in many cases, the problem that users have with products, and especially with documentation, is that the information is presented only at a very detailed, low level of the task. The user is looking for a way to meet a goal and the manual doesn't have any headings that relate to the user's goal. The documentation assumes that users know which tasks to do to reach their goals.

For example, if the product has a function for "grouping objects," the help is likely to have a topic on "grouping objects." For users who know about grouping, that's fine. But for users who don't already know that "grouping" is the way to move the parts of a picture together that may not be fine. They may not pick up on "grouping" as the task they want to do. Even if they see an icon, tool tip, or menu choice for "grouping, "they may not select it. Those users are looking for an entry on "keeping the parts of the picture together when I move it." The goal is to move the picture without having to move each piece and reposition it in relation to the other pieces. "Grouping objects" is a task the program makes users do in order to achieve the goal. The manual might have a section on "working with several objects or parts-grouping" that connects the novice's words with the program's name for the action. The index in the manual and the search list in help should have entries for "grouping" and also for perhaps "moving-part of the pictures together" or other phrases users look for. You find out if users have those problems by watching them do similar tasks and listening for their problems and their words for what they are trying to do. To design successful interfaces and plan successful

documentation, you have to do task analysis on all levels from understanding the users' goals to knowing the tasks they
do to achieve those goals to understanding how they now carry out those tasks.

Identifying different types and levels of task analysis

Goals are a critical part of task analysis. So is understanding what users do to meet those goals. In planning a task
analysis, you might be interested in one or more of these variations on the theme, which we discuss in the next several
sections of this chapter: how work gets done when several people are involved (workflow analysis)

- what a single individual does throughout the day or week or month (job analysis)
- how workflow analysis interacts with job analysis
- what tasks are performed by all the people who might be using your product (task lists or task inventories)
- the order in which users do tasks (process analysis, task sequences)
- how a large task is made up of subtasks (task hierarchies)
- what steps and decisions users take to accomplish a task or part of a task (procedural analysis)

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Workflow analysis

Early in a project, you may want to understand how a particular process is accomplished even if several people are
involved in completing that work. This is "workflow analysis" or "business process analysis." Many companies are trying to
simplify business processes. To do that you first have to understand the current process and then look for redundancies
and unnecessary steps in that process.

*Consider the process of getting a prescription refilled. At least two people are involved: the patient and the
pharmacist. More people may be involved, including a relative, friend, or other caregiver of the patient; a
clerk or pharmacist's assistant at the pharmacy; a receptionist at the doctor's office; and a doctor. In fact, if more refills
are not already allowed on the prescription, the pharmacist must contact the doctor before refilling the prescription. A
workflow analysis would reveal that, in a typical situation.*

1. The patient contacts the pharmacy.
2. The pharmacist or pharmacy clerk takes the information.
3. The pharmacist looks up the patient's prescription and realizes that approval is needed for a refill.
4. The pharmacist calls the doctor's office.
5. The doctor's receptionist sends the call to the doctor or takes a message for the doctor to call back.
6. The pharmacist waits for a call back.
***7. After the call back, which the pharmacy clerk takes, the pharmacist fills the prescription or contacts the patient
to say the prescription can no longer be filled.***

The work flows across people, and in this case across sites, as shown in figure 3-4.

Figure 3-4 You may want to see how work flows across people.

Workflow analysis is a very important type of task analysis because the situation in which different types of people are
involved in the process is much more common than processes individuals do alone. The people involved in the process
may be at different sites, as in our example of pharmacists and doctors. They may be in different divisions within a

company. For example, getting an invoice paid may involve a technical person who receives the invoice, various levels of
management who must sign off on it, the accounting clerk who now puts it into the computer, another clerk who pulls up
all the invoices due to be paid on a certain date and prints the checks, and a mail room clerk who stuffs the checks into
envelopes and puts on postage. Or all the people involved in a workflow may be in the same division or office.
If you do a task analysis by looking only at one part of a workflow process, you risk developing a product that will not be
used because it is incompatible with the rest of the workflow.
To do a workflow analysis ofgettinga prescription refill approved, you would need to spend time in both
pharmacies and doctors' offices. You would need to observe and talk with pharmacists and their clerks. You
would need to observe and talk with doctors and their receptionists. You might also want to talk with patients and
their relatives, friends, and other caregivers.
Interviews and observations with the people involved might reveal that up to 80% of the calls that come into a
doctor's office in the course of a day are for approvals on prescription refills. They may also reveal that the calls
interrupt what the receptionist and the doctor are doing and that the calls back to the pharmacist interrupt the
pharmacy clerk and pharmacist. That's all part of the workflow analysis. You might conclude from the workflow
analysis that the current process of getting approvals for refills wastes time that the people doing the work would
rather spend doing other tasks: filling other prescriptions in the case of the pharmacist and seeing patients in the
case of the doctor. You might conclude that opportunities exist for simplifying and streamlining the process. What
if pharmacists and doctors could communicate by computer? Pharmacists could send messages and check for
Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish
replies (from many doctors) at times convenient for them, and doctors could check for messages (from many
pharmacists) and send replies at times convenient for them.
In most cases you will be doing site visits because you already suspect that a problem exists and you already have an
idea for solving the problem. In gathering data for your workflow analysis, you need to focus on getting information that
will help you decide on the probability that your idea will succeed with the users, help them meet their goals, and support
their values. You want data that will tell you what you must do in the design to make your solution attractive to the users.
You want data that will tell you about constraints that you need to take into account in your design.
In the example we've been using about getting prescription refills approved, part of your workflow analysis might
be to find out who actually does the calling back and forth. If, as often happens in workflow analysis, the answer
isn't "always person A, " you should get information on frequencies. How often is it the pharmacist who calls?

How often is it the clerk? How often is it the doctor who actually returns the information? How often is it the receptionist who actually talks to the pharmacy?

Note that to get a good sense of these frequencies you would need to select the sites to visit to give you the best chance of observing the range of possibilities. That is, you would probably want to go to some small pharmacies and some large ones; some individual medical practices and some large practices with several doctors and a larger staff.

Why is knowing how the work flows today important? You might think that it is most likely that the person who now makes the calls will be the person who is most likely to put the information into the computer and that the person who now takes the calls is most likely to take the information from the computer. However, you might also be thinking about changing the tasks that people do or the roles that people play in the process, eliminating some and moving others from one person to another. As you do your workflow analysis, you should consider and make notes about each user's goals and values as well as the environment to help you decide if the changes you are contemplating are going to be acceptable to the users.

In our pharmacy and doctor workflow example, if you are thinking of changing from using the telephone to the computer for refill approvals, you will need to consider issues like, Where are computers in doctors' offices and in pharmacies? What is the likelihood that the environment will change? Do you expect the doctor to use the computer? How likely is that?

Although we've separated our discussion into chapters on users, tasks, and environments, during *a site visit you are* considering all *three at the same time.*

So even though the computer doesn't figure in the actual workflow of the current process in our example, if you were doing observations and interviews to gather data on the current process, you would also want to look at and ask about computer use: Do the people involved now use computers? What do they use them for? How comfortable are they on the computer? We know from other studies that doctors don't like to look as if they can't do something. So if the system you design isn't easy to learn and use, the doctor might not be the person who actually looks for and replies to requests for refill approvals. When the requests come over the computer, the receptionist might receive them and a paper system might develop between the receptionist and the doctor that they perceive as more cumbersome than the current system of phone calls. Your idea of changing the workflow might be defeated.

In a workflow analysis, you should also extend the workflow to look not only at the people who are involved in the actual work you want to change but at the people who are involved at the entry and exit points of the workflow. Their goals and values may impinge on what you can do to improve the workflow and on the way you design new processes.

In our example about getting approvals for refilling prescriptions, you would also want to consider the patients'

and caregivers' goals. How much time on average does it take in the current system to accomplish the task of
getting the prescription refill approved no matter how much it interrupts or bothers the pharmacist and the doctor?
If the patient needs the prescription quickly and the turnaround time in the new system is longer because doctors
and pharmacists only check the system at intervals during the day, the patient may be less pleased and prefer a
pharmacist and doctor who work in the old-fashioned but immediate response, phone call mode. If a typical
scenario is that a caregiver comes to the pharmacy thinking the prescription can be refilled, finds out it can't, and
now waits for the approval to come, the caregiver is likely to be extremely anxious about time. The caregiver may
have left the patient at home and perhaps would find it very difficult to make another trip to get the prescription
later. In designing a new system, you want to take these needs into account.

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Workflow analysis shows who does what at each step of the process. In our view, however, workflow analysis is more
than drawing flowcharts of the steps in the process and showing who does each step. It includes understanding the goals
of all the people involved in the process (what they value). It includes understanding them as users, especially in relation
to technology or other changes you are thinking of making to the workflow.

Job analysis

Another type *of task analysis you might want to do early in a project is job analysis-understanding all the work that a*
person in a certain position does during the day or week or *month. If workflow analysis is a horizontal picture of how work*
moves across people, job analysis is a vertical picture of all the types of work that flow through a particular person as
shown in figure 3-5.

Figure 3-5 You *may want to see all the work that a particular person does.*

Job analysis can help you

• **find** new marketing and development opportunities (What are these people doing for which your company
might develop new products that would make their jobs easier?)
• **understand** specific features to build into your product (If these people get interrupted a lot while working,
you may need to consider how to help them get into, out of, and back into your product quickly and easily.)
• **learn** what pressures they are under and what they value (If typical users do the same task repeatedly, you
will want to develop and test your product in a different way than if typical users do many different tasks or
will only use your product occasionally.)

Although we've called this type of task analysis "job analysis," it really means what people are doing over a period of time
and across different activities. You might, in fact, be interested in what people do at home, in their personal lives, or for
leisure, instead of what you might have thought of as their "jobs." Susan Dray and Deborah Mraczek visited families in the
United States, France, and Germany for Hewlett-Packard to understand how families use computers (Dray and Mraczek

1996a, 1996b). A Microsoft team visited families to "build a comprehensive understanding of home activities and how they
are accomplished, systematically examine home activities for new software opportunities, and determine ways to improve
usability and increase customer satisfaction with Microsoft's consumer software products" (Juhl 1996, 215).
To do a job analysis, you must work directly with the people in that job. Written job descriptions seldom give you a realistic
view of what people in a job actually do during the day. Even managers telling you about the people they supervise may
not give you the full picture of what these people actually do and how they accomplish their tasks.
You should actually observe and talk with the people as they do their jobs or have them keep track of what they do for
you. Just talking to them outside of the context of the work (or leisure) may not be enough. People don't remember what
they do all day. Away from the situation, their sense of how much time they spend doing something or how often they do it
may not match reality.
The best way to do a job analysis is to spend time with people on the job. In different companies, this is called
"shadowing," "walking in someone else's shoes," "walking a mile with someone," or "a day in the life of..." If you cannot
spend time with users, you might have users keep a diary of what they do, writing something down every fifteen minutes
or checking off items from a list you've built from previous observations. (See the section on the technique of "customer
partnering" in chapter 6 for more ideas on having users keep track of what they do when you aren't observing them.)
Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish
In doing this type of task analysis, you might be interested not only in the list of tasks that people do in a given job, but
also in factors such as
• **Frequency.** How often do these people do each task?
• **Criticality.** How important is each task to their job?
• **Time to complete.** How time consuming is each task?
• **Difficulty.** How much of a problem is accomplishing this task?
• **Division of responsibility.** Do all the people in that job do this task? Do different people with the same job title do
different tasks?
As with any type of task analysis, you want to watch, listen to, and talk with several users with varying degrees of
experience and responsibility to get a good overall picture of the job you are interested in.

Combining workflow analysis and job analysis

An overall task analysis might include both workflow analysis and job analysis.
*Let's say managers in your company complain about how much time their administrative assistants spend on the
phone making arrangements. You do a job analysis, sitting with administrative assistants, talking with them, and
having them keep diaries of what they do. You find that one of the most frequent, critical, and timeconsuming
tasks they do is arrange meetings. You then have several administrative assistants talk you through how they
arrange meetings and you sit with one administrative assistant for an entire day. You find that first they get*

information on the times when all the potential participants are free. This usually requires several phone calls,
often with time spent waiting for calls to be returned. Then they try to find a room for the best times by calling
facilities management. That also requires at least one phone call, sometimes several. Based on this job analysis,
you conclude that an automated scheduling program could make the administrative assistants' lives much easier
and their bosses happier.

If you design a new program just for the one group you watched and talked with, however, it may not have as widespread
use or as much value as it could. You should also ask: Who else in the company could use this program? You are really
asking: Who else does a similar task? You are saying: What other job analyses should I do?

In our example, you would ask: Who else schedules meetings? Who else wants to use the rooms that the
administrative assistants are booking for meetings? You would probably come up with several other
types of people in the organization, from staff who do their own scheduling because they don't have administrative
assistants to people in specific divisions who need large rooms for their work, such as trainers. You might now
want to watch and talk with people in those other jobs. How do they schedule meetings and training sessions?
Would they need different features in an automated scheduling program? What words do they use for the tasks
they might do in the new program? What difficulties do they have with the way the task is done today that you
could also overcome in a new program?

In addition to thinking about other people who do the same task, you should also think about the flow of work that is
required to have the overall work you are concerned about go smoothly.

In our example of an automated scheduling program, you would now ask: Who else would have to put information
into an automated scheduling program or get information out of it? You might think of facilities staff who are in
charge of the rooms, audio-visual staff who are in charge of overhead projectors, VCRs, etc., and cafeteria staff
who bring refreshments to meetings. (See figure 3-6.)
Within each of these groups, someone might be in charge of putting information about options that schedulers can
select into the database and changing the information in the database when needed. For example, someone in
facilities would probably be in charge of keeping the list of available rooms up
to date, indicating for each how many people it holds and what different choices meeting schedulers can select for
the physical arrangements of chairs and tables. Other people in each of these groups would need to be able to get
information out of the program. For example, people in the cafeteria might use the program to know which
meetings have arranged for refreshments, what refreshments, and
what time to deliver them.

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Figure 3-6 You may want to do both a workflow analysis and several job analyses with the people involved in the workflow.

If other users' tasks related to the work aren't considered and included in the program, the person whose job you were
trying to improve might still have to do a lot of work in the old way.
In our example, *if* facilities, audio-visual, *and cafeteria staff* tasks aren't included as you design the automated
scheduling program, the administrative assistants will still need to make many phone calls. *If the* tables aren't in
the right place, if the *overhead projectors aren't* delivered, if *the coffee service doesn't come on* time because
users in facilities, AV, *and* the cafeteria can't use the program easily, the entire project to automate *scheduling*
might fail.
Workflow analysis helps you to find all the user groups you need to involve, and it helps you understand all the tasks that
have to be included in the task list for the product. Even if you cannot put all these functions into the product right away,
you can design to avoid conflict with them and you can design to phase them in later and have them fit smoothly into the
product. If you know about all these users from the beginning, you can make design decisions that will work for them-even
if the functions that focus on them aren't in the first release.

Task analysis to develop a task list or task inventory

Part of the predesign analysis you must do for any product is to develop a task list. A task list is an inventory of all the
tasks that users want to accomplish within the overall area that your product will handle.
You might at this point ask the question "What is a task?" You might say, "I know it's what someone does. I know you've
said it's what someone does to achieve a goal. But how do I know if I'm listing my tasks at the right level?" What you are
really talking about is granularity: How big a task should you think of as a task? The question "What is a task?" came up
on the listserve and one of the usability specialists on the listserve gave the following very clear and useful reply:
"I'd suggest a task is any observable, measurable action that has an observable beginning and an observable end. For
the sake of consistency, you may want to agree among yourselves [a project or design team] about the level of granularity
you'll use. For example, is the task "prepare dinner" or "peel the potatoes"? What you decide is a job, a task, or a step
doesn't really matter, so long as you're consistent among yourselves. It probably would also be worthwhile to think about
how long your list would be using whatever level of granularity you decide on: a list of five tasks or of five thousand tasks
may not be helpful." (Dick Miller, listserve contribution, July 1997, used with the author's permission.)
When getting an initial task list to decide on a product's functionality, you might, in fact, want to start with a relatively short
list of tasks: the ten to twenty major functions that users will want or need to do with the product you are designing. Figure
3-7 shows you at least part of such a list for an electronic mail program.

Parial task list for an e-mail program

- **Write a message**
- **Send a message**
- **Receive a message**
- **Read a message that you received**
- **Reply to a message**
- **Save a message to look at it later**

• Forward a massage to someone else
• Send a formatted file with the message
• Send the same messge to several people
Figure 3-7 Part of a task list for an electronic mail program.

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Later as you plan the design, you'll want to break those tasks down into smaller units at least until you get to the level that is likely to match the menu choices and icons of an interface, the titles of help topics, and the second or third levels of headings in a user's manual. For example the high-level task of "Keep an address book" might include lower-level tasks such as

• Start a new address book
• Add someone to the address book
• Change information about a person in the address book
• Delete someone from the address book

A task list does not tell you how the users will accomplish the task. It only indicates what the users have to be able to accomplish with your product. The how is your design problem. The goal of design is to figure out the best way to create a product that lets users easily and quickly do these tasks.

A useful task list is one that lists all the tasks that all the different types of users want to accomplish, and names the tasks from the users' point of view and in the users' words. Thus, in getting a task list for an e-mail program, you would probably have a task of "sending the same message to several people." Put it in your task list that way and not as "using the Group Distribution function."

This users' task list should drive decisions on the product's functionality. If the point of a product is to help people do work, the functionality of the product should come from understanding the overall work and the specific tasks that users need to accomplish-not just from the capabilities of the technology.

You may, of course, add tasks to the list that users don't do now or that they would not think to put on the list because they don't know that they will be able to do them. That's often the advantage that you get by introducing new technology a that users will be able to do tasks they can't do now. However, the new tasks must still meet goals and needs that users have.

In many cases, tasks that you think of as new aren't really. For example, you might "think that "setting up a distribution list" and "sending the same message to several people" are new tasks for an e-mail program. In fact, people were doing these tasks before, just in different ways. They were putting distribution lists on memos, amaking many copies of them, and distributing the copies.

You get a task list by observing and listening to users doing similar work, by talking with users either individually or in groups about the work they do, and by collecting scenarios or stories of their work. As with job analysis, issues of frequency, criticality, time to complete, difficulty, and division of responsibility are relevant to gathering a task list.

Process analysis, task sequences

A task sequence is a series of tasks on the list that users must do or are likely to do in a certain order, as shown in figure
3-8.

In an e-mail program these sets of tasks have a natural sequence

- **Write a message Come before**
- **Send a message**
- **Receive a messagse from some one else Comes before**
- **Reply to a message, or**
- **Forward a message to someone else**

Figure 3-8 Two natural task sequences for an e-mail program.

The task sequence may be logically necessary: B cannot be done until A is complete. You cannot send a message until it
is written. You cannot reply to a message or forward the message until you receive it.

However, the task sequences that you observe in site visits may not be logically necessary. Users may be doing tasks in
a certain order just because they've always done them that way. If you find everyone doing a sequence of tasks in more
or less the same order, it may be because the current version of the product leads them to do it that way, because
management has imposed that order to achieve consistency across users in the way that things are done (standard
operating procedures), or because users have been trained to do the tasks in that order.

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

You may also find that different users do the same tasks in a different order. Even if supervisors tell you that there is a
prescribed task sequence and everyone does the tasks in the same way, you may find individual differences as you
observe. Except in highly prescribed fields, usually where safety is a paramount issue and where users have extensive
training, you will generally find many variations on task sequences even when supervisors tell you everyone is doing the
same thing.

You can gather information for task sequences from observations and from interviews. Getting information on task
sequences is a process analysis just like workflow analysis. The difference is that in workflow analysis you are getting
information on how a process is done that involves different users at different points in the process. For the type of task
sequence we are discussing in this section, you are getting information on how individual users each do a process, as
shown in figure 3-9.

Figure 3-9 Doing a process analysis of two different people sending a letter to someone. Note the individual differences in the task sequences.

You can get information on task sequences with the same methods you use to get task lists. In particular, when you
observe users, note the order in which they do the tasks to meet a goal. As you gather information about task sequences,
try to find out if users always do the tasks in that order and why? If different users do the same tasks in different orders or
if one user does it one way sometimes and another way at other times, try to understand how important that flexibility is to
the users.

When you begin designing the new interface, make use of this information. Consider building in flexibility to meet
individual differences. For example, if you find that users sometimes write messages before addressing them and

sometimes address them first, don't create a software product that forces users to select an address before composing
the message.
You may find that typical current task sequences aren't very efficient. Much of the work in reengineering business
processes is changing task sequences to make them more efficient. That can help everyone, but you should also be
aware that many users, especially users who have been doing the same task sequence over and over for a long time, will
find it difficult to change. Don't arbitrarily change task sequences without good reason. When you do change task
sequences, you should think about how you are going to help users make the transition to the new process.

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Task hierarchies

Task analysis is hierarchical. We can take tasks at any level and decompose themdivide them up into pieces-to see the
tasks at greater and greater levels of detail. We can take a job and divide it into the tasks that make up the job. We can
take one large task and divide it into the smaller tasks that make up that task (see figure 3-10). We can take even a
smaller task and show all the steps and decisions in it (see figure 3-11).

Figure 3-10 Task analysis is hierarchical. You can breakup a job into tasks and each task into subtasks .

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Figure 3-11 A task list may include tasks at different levels. Be aware of the relationships. This figure includes four tasks at the same level and then the procedure (subtasks
and decisions) for one of the four main tasks.

That's the issue of granularity. As you think about task analysis and your predesign site visits, you need to decide what
level or levels you are most interested in and how deeply you want to decompose each task. When you gather information
in your site visits about users' tasks, you may find that you have put tasks at different levels of granularity into the same
task list. You'll want to look through your list and perhaps group some of the tasks that belong together under the same
larger task.

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Procedural analysis

When we take one specific task and divide it into the steps and decisions that a user goes through in doing that task we are doing "procedural
analysis" (see figures 3-11 and 3-12).

Figure 3-12 You can carry a task analysis down to the individual steps and decisions users make as they carry out the task.

The procedural level differs from the types of task analysis we have been talking about up to now because it always
includes working within the constraints of a specific interface. You are seeing how users carry out their tasks with the tools
they currently have.
You may be planning to entirely change the users' procedures. Nonetheless, it may be very important to get a detailed
picture of just how users actually do the tasks today. A procedural analysis allows you to understand both the physical
steps that users take and the mental decisions they make while doing a specific task. Doing a procedural analysis of
users working in their current mode may give you important insights for your new design.

You do a procedural analysis by observing and listening to real users doing real tasks. Although you can do this in a
usability laboratory, you get another dimension of reality by watching and listening to them in their real environment as
you do in site visits.
Depending on your objectives, you can use your procedural analysis to create an idealized picture of the process that
typical users typically follow. Or you can show an actual flowchart of the process as a particular user did it with all the
errors the user made and the wrong paths and repeated loops that the user took as shown in figure 3-13. Procedural
analyses of real users doing real work are often quite messy, but they are very informative. They show where current
processes are difficult for users, thus providing opportunities to improve the process and to improve the software and
documentation for doing the process.
To recap what we've covered, task analysis means understanding the work users do and how they do it. With your
understanding of users' goals, you can look at work in many ways and at many levels, including workflow analysis, job
analysis, combining workflow and job analyses, task lists, task sequences, task hierarchies, and procedural analysis.

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Thinking of users according to their stages of use

In this last section, we relate what we covered in chapter 2 about users to what we covered in the first two parts of this
chapter about goals and tasks. In addition to all the ways of characterizing users that we discussed in chapter 2, users
differ in their interactions with a product because of the specific tasks they do and the frequency and expertise with which
they use the product. As we think of users in terms of frequency and expertise, we can characterize them as being in
different "stages of use," from novice to expert.
The model of characterizing users by stages of use was originally developed by Dreyfus and Dreyfus (1986) in minds
overmachines. We have expanded the model through observations that we and others have done during usability testing
and site visits and through the work of Scandinavian researchers and others who have focused on the collaboration
between users and designers for the development of software interfaces. (See, for example, Elm 1989, 1993).

Figure 3-13 You can use a task analysis to show how a real user really does a task. This is a middle-aged man in his living room trying to program his VCR, a task he has
once done with another VCR but has never tried with this one.

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Any particular user at any particular moment in time with any particular product is at one of the four stages of use:

- novice
- advanced
- beginner competent performer
- expert

Although with any particular product you will find some users who progress through all the stages, most don't go all the

way to expert performance. Nor do they need to. We cannot all be expert at everything. We all "satisfice" (Simon 1976) by trading off time and effort for the benefit of greater expertise. Depending on our need, we may remain more or less a novice with some products, be an advanced beginner with many, and be an expert in other aspects of our lives.

The stages of use provide a useful way of categorizing users whom we observe in field studies, as well as a model for thinking about interface and information designs. Some interfaces and information products, for example, are designed primarily for advanced beginners who will perform a few functions over and over again. Other interfaces and information products are focused more exclusively on the needs of competent to expert performers, those who have considerable previous experience with similar products and who want to use the new products as comprehensively as possible. Some must meet the needs of users over the entire range.

Users change over time

Although not everyone becomes an expert, one of the most important points that we gain from the stages of use model is the reminder that users do change over time. Although every user is a novice initially, most move on to being advanced beginners. Many may want to be competent performers. And some users, given sufficient motivation and time, transform themselves into experts. Facilitating movement through the stages should be one goal of design-giving users a product interface and information that supports their continued learning and experimentation. In fact, if we design an interface effectively enough, our most experienced users will find things to do and ways to do them that have never occurred to us.

Occasionally we hear developers argue that they need not study potential users because the users will change once they must interact with a new product. Certainly users change as they add skills and knowledge through their experiences with a product. However, products will succeed only if they facilitate users having successful first experiences, and only if they also allow for growth and learning and for a variety of patterns of use. To design those successful products, you must understand the users' starting points and how users learn and change over time.

Let's look at each of the four stages of use to understand the characteristics that are typical of users at each stage.

Novices

All new users of a product are novices at first-they do not know exactly what to do to use the interface or information. It may be similar to something they are already comfortable using, or it may be drastically different. It may do one or many things that they have been unable to do before, or at least not so easily or conveniently. Figure 3-14 highlights typical characteristics of novice users.

Novices are often faced with a great, and sometimes frightening, unknown. What will happen if they fail to learn and perform effectively? Do they risk embarrassement? Loss of a job? Frustration? Anger? Or simple annoyance? No one likes to look foolsh, even to one's self

- **fear of failure, fear of the unknown**
- **Focus on accomplishing real work**

• **Impatient with learning concepts rather than performing tasks**
• **Theorical understanding only – no pratical experience**

Figure 3-14 Characteristics of novice users.

Luckily, in some instances, being a novice can be a short-lived experience. Sometimes the novice experience lasts only minutes or hours as the user becomes comfortable with the differences between the old known behaviors and the new unknown ones. Approaching a new product armed with prior knowledge of and experience with similar products doesn't eliminate novice behaviors or feelings, but it may considerably shorten their duration. The better the product is designed to account for users' prior knowledge and experience, the .shorter and more pleasant the novice experience is likely to be.

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

However, in other situations, being a novice can be extremely unpleasant. If the product is poorly designed, novices may find themselves completely stumped, unable to succeed, frustrated at wasting precious time without achieving a goal. The new behaviors necessary to succeed in using the product may seem awkward or frustratingly different from previous experience. Novice users, faced with a poorly designed new product, one that does not accommodate their goals, may react with an extreme fear of looking stupid. They want to succeed, they want to perform some task successfully, they want to achieve a goal, but they are unable to do so. If they fail, they may put the product aside "for another time." They may give up and never use the product. They may even choose to leave their jobs if using the new product is an absolute requirement.

From observing many novices approaching new products, we can offer two critical generalizations to the designer:

• Novices are very goal and task oriented.
• Novices often do not want to learn, but simply to do.

Novices are often impatient to accomplish some task without taking time to learn or develop sound mental models of the new product. Novice users of an information kiosk in a museum are not interested in becoming skilled at using the kiosk interface. They want to browse the information or find a specific answer immediately. Novice users of a desktop publishing program want to start creating pages of text, not first become experts on the conceptual differences between word processing and desktop publishing.

All users will be novices with your new product when they first get it. But not all novices are the same. As we said in chapter 2, no user is a "blank slate:" All are nonnovices in other ways. All bring their prior experiences and knowledge to their use of your new product. As you work with potential users on site visits, you need to find out about their prior knowledge and experience.

You need to gain a good sense of where users are in terms of subject matter or domain knowledge, experience with the technology, and experience with previous versions of the product. For example, if you are designing a new Windows version of a hotel reservation system that has existed as a DOS program, you want to know:

• How much do typical users know about making hotel reservations? (That is, subject matter or domain
knowledge)
• How familiar and comfortable are users with Windows? (That is, experience with the technology.)
• How much do users know about the old DOS-based hotel reservation program and how long have they used
it? (That is, experience with previous versions of the products.)
Will there be users who have never made hotel reservations, who are unfamiliar with Windows, and who have never used
the old program? If turnover in hotels is high, you may have people who are novices in all these ways. Will there be users
who know about making hotel reservations and have used the old program but who will be new to Windows as well as
your new version? They are novices in some ways but may be experts in the subject matter.
Novices who are subject-matter experts are likely to transfer their previous experience. If most new users will be
transferring from an earlier way of doing the same process, even on paper, you must understand that current process and
build the new program and information materials to facilitate the transfer. Consider how your design will accommodate
these users' mental model of the tasks. Novices who are not subject-matter experts may not be loaded down with that
previous knowledge and experience, but will then have to learn to make a hotel reservation while they are also learning a
new program.
Novices who are competent or expert performers in their subject matter and already comfortable with the technology are
more likely to move quickly from stage one (novice) to stage two (advanced beginner). When the subject matter is also
new, however, even a well-structured interface may initially seem extraordinarily difficult to learn. Learning both a new
technology (for example, Windows) and a new program (for example, a reservation system in which the order of doing
tasks has been changed) at the same time can be overwhelming. If many users will be in that situation, consider including
interface assistants (wizards, cue cards, coaches, and guides) and teaching tools, such as print manuals, tutorials (both
paper and computer-based), and online help in your design.
You also need to evaluate how long most users are likely to remain as novices. How often will the same people use what
you are developing? How much turnover will there be among users? How much do users want to invest in learning?
Users of walk-up kiosks are nearly always novices. Some definitely gain experience, but kiosk designers should not count
on having expert users as they design. On the other hand, computer programmers using a new version of a language

compiler are likely to move quickly through the novice stage and prefer a product design that quickly accommodates
higher stages of use.
In designing your site visits, consider the different types of novices you will find. You should observe some people who are
new to the subject matter, the technology, *and* the product. You should observe some people who are transferring to your

product, including a range of people who are at different stages of use with the technology and with older versions of the
product. In many site visits, we make a particular effort to observe users who are novices at performing the tasks
(subject-matter novices). We also make a particular effort to observe advanced beginners with the current process and
product, not just experts who are usually far less typical users.
In your site visits, also pay particular attention to frequency of use. Infrequent users may always behave as novices.
Frequent users are more likely to move quickly from novice to advanced beginner.

Advanced beginners

Once novices move beyond the fear of failure and begin performing the tasks that allow them to achieve their goals, they
become advanced beginners. An advanced beginner is a user who is focused simply and exclusively on getting a job
done as painlessly and quickly as possible. In most instances, advanced beginners use a particular product only
incidentally and infrequently in their work or home lives. Very few of us, for example, choose to be any more than
advanced beginners of our microwave ovens. We are content to heat leftovers, boil water, or use the builtin timer. The
microwave allows us to perform many more exotic tasks which we have neither the inclination nor the time to learn. We
learn to answer and make calls with the telephones on our desks; we may even become adept at forwarding calls or
placing them on hold. But how many of us have to look up the instructions for arranging a conference call, or better yet,
enlist the help of the local expert to do the arranging for us? We're not interested enough in the intricacies of the phone
system to learn more than a few basic tasks.
Even in the workplace, people use many products, especially application software, only incidentally and infrequently as
part of their jobs. They may use desktop publishing software to write an occasional letter or memo or a spreadsheet
program to set up a small table and perform a calculation. They are quite content to learn a few tasks they use regularly,
such as setting selected words in bold type or using the summation function to add rows and columns, but they're unlikely
to learn about creating style sheets or using the spreadsheet to perform statistical analyses of data. Figure 3-15 highlights
typical characteristics of advanced beginners.

- **Focus on accomplishing real work**
- **Impatient with learning concepts rather than performing tasks**
- **Randomly access tasks**
- **By adding new and progressively more complicated tasks, begin to develop an empirically bases mental model**

Figure 3-15 Characteristics of advanced beginners.

Many users never learn to perform more than a few tasks with a product. In fact, up to 80% of the users of a typical
software or hardware product never move beyond the advanced beginner stage of use. By definition, higher stages of use
require experience with many tasks, often over longer periods of time. Thus users who are content with a few tasks
remain advanced beginners.

Many advanced beginners, focused on performing the tasks they need, are content to ignore the rest of the interface. If, however, we have not, as designers, recognized the needs of advanced beginners and identified the most often performed functions, we may create an interface that is difficult for advanced beginners to navigate. For example, interfaces that provide too many alternative methods of performing the same function may confuse advanced beginners.

Interfaces that emphasize exotic, and infrequently performed, functions may obscure frequently performed functions.

Interfaces that fail to optimize the most commonly performed functions will be difficult for advanced beginners to learn and use.

Like novices, advanced beginners are often reluctant to spend precious taskoriented time learning concepts. They tell us that they don't have time to learn, only to get on with the job. However, as advanced beginners gradually increase the number of tasks they can perform, they begin to form a concept or mental model of how the software or hardware interface is organized. This mental model enables them to learn new tasks more easily, especially if the interface they are using is unswervingly consistent. For advanced beginners to learn effectively, a consistent interface is a necessity.

Consistency allows users to plan how they might perform a new task based on their experience performing previous

tasks. Their plan relies on their mental model of the task. If the interface has maintained consistency among tasks, the users' plan will work, and they will be successful in learning the new task.

Advanced beginners who are already skilled in the subject matter but not in the technology are most likely to be impatient learners. They will try to make sense of the interface to accomplish their goals without assistance from documentation, tutorials, or interface assistants. Only when they cannot make sense of the interface will they seek outside assistance from people who already know what to do, technical support services provided by the manufacturer, or the documentation, whether online or in print. They are also most willing to spend time learning to use the new product because they will see the greatest pay off from the time and effort. Subject-matter experts, or those who need or want to become experts, are likely to continue learning new tasks, moving into the next stage of use- competent performance.

On the other hand, advanced beginners just learning the subject matter are likely to concentrate on learning the few simple tasks that they need to perform at the moment. And advanced beginners who are infrequent users and who do not see themselves as ever becoming subject-matter experts are also likely to learn only the tasks they need to reach their immediate goals.

The difference between novices and advanced beginners is that the advanced beginners can perform several tasks well.

The difference between advanced beginners and users at higher stages of use is in part the number and complexity of the

tasks they have learned and, in part, the advanced beginners' difficulty in handling problems. Advanced beginners are usually not comfortable trying to diagnose and correct problems and are often unsuccessful at it. In your site visits, you might want to pay particular attention to signs of confusion and requests for assistance when advanced beginners encounter problems.

Differences in frequency of use, breadth of use, and subject-matter knowledge also distinguish advanced beginners from users at higher stages of use. Infrequent users of your products and information are unlikely to progress beyond the advanced beginner stage. Frequent users, even ones who work with the product all day every day, may remain advanced beginners if they only do one or two tasks repeatedly and have little understanding or interest in the broader domain that the product covers, and may remain uncomfortable with anything beyond the few tasks they know well. Users who lack subject-matter knowledge are likely to remain advanced beginners longer than those who are competent in the subject matter. As you plan your site visits, decide how you will identify the mix of advanced beginners and users at higher stages of use. Look for situations and users who differ in frequency of use, breadth of use, and subject-matter knowledge.

Don't rely on users characterizing themselves on scales in which you ask them to rate how comfortable or how knowledgeable they are with the subject matter, the technology, or the product. Advanced beginners are likely to rate themselves as more comfortable and more knowledgeable than users at higher stages of use. They are comfortable with what they do, and they don't know what they don't know. More competent performers are often more aware of their lack of knowledge.

Competent performers

In the stages-of-use model, competent performers are defined as those users who have learned a sufficient number of tasks that they have formed a sound mental model of the subject matter and the product. Competence, as a result, comes only with experience, not from first learning a conceptual framework for the tasks.

As competent users gain experience using a product to perform more tasks, they begin to understand how the various tasks fit into a whole. As a consequence, they become better at foreseeing how an interface will behave and planning how they will perform a new task. They also increase in their ability to diagnose and correct problems. Novices and advanced beginners are often completely stymied by unexpected results of their actions. They have few, if any, resources to trace back what they have done and recognize actions that did not produce the results they wanted. Competent performers, because of their increased experience using the interface, become much more skilled at recognizing an incorrect series of actions and finding ways to trace back and correct them. Figure 3-16 highlights the typical characteristics of competent performers.

- **Focus on performing more complex tasks that requere many coordinated actions**
- **Ability to plan how to perform a complex series of tasks**

- **Willingness to learn how tasks fit into a consistent mental model of the interface as a whole.**
- **Interest in solving simple problems by applying a conceptual framework to diagnose and correct errors**

Figure 3-16 Characteristics of competent performers.

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

In a usability test, JoAnn watched two users struggle with setting up the fax application for their new computer system. Both were competent users of *fax machines, understood the concepts behind sending faxes from a computer, and were skilled users of the Windows 95 operating system. However, as new users of the Windows 95-based fax product, they were confused about the sequence of events required to set up the fax system and send a fax. They finally got the fax set up and ready for use but only after a confusing sequence of trial-and-error attempts. At the end of the usability session, one of the users commented that he "had no idea how [he] did that. "*

A few weeks later, however, after they had installed the fax product once again and used it to send and receive faxes, these users had developed a comprehensive mental model of the way to do tasks in the new product. They commented that they could easily set up the system again because they now understood that they had to create a folder to receive a fax before they could complete the set-up process. They understood the new process even though they continued to view the required sequence as somewhat peculiar. In fact, the required sequence continued to violate their previous mental model of the task. They asked, "Why must 1 have a folder set up to receive faxes, when I only want to send a fax?" However, they now knew they had to do it. In the process of gaining competence at using the product, they had acquired a conceptual understanding that would help them diagnose and correct problems using the software in the future.

In observing competent computer repair technicians, we noted that they were frequently able to diagnose problems on products entirely new to them because they were able to apply a sophisticated mental model of how similar computers worked. They were able to distinguish between significant and insignificant differences among products.

Competent performers are more likely to use job aids provided with an interface to increase their understanding of how the product works. They are often anxious to learn additional tasks and have sufficient understanding of the tool to understand detailed task-oriented instructions. Competent performers with considerable subject-matter knowledge and experience using similar interfaces are also more likely to use tutorials and computer-based training to quickly gain familiarity with common tasks and the conceptual model designed into the product. Since competent performers are also involved in diagnosis and error correction, they are also more likely to use documentation to gain insights into how things

are supposed to work and what might have gone wrong. They are likely to prefer, however, straightforward
problem-solution tables to more conceptual discussions of the design and the theories behind the design.
The ability to create a plan of action to perform a task and follow it to a successful conclusion, plus the ability to recognize,
diagnose, and solve problems while attempting to perform a task are the hallmarks of a competent performer. When you
conduct your field studies, try to discover what percentage of your users have achieved competence by observing their
ability to perform complex tasks and to solve problems. In distinguishing between competent performers and advanced
beginners, look for the diversity of tasks performed. Advanced beginners are likely to know how to perform fewer tasks
overall than are competent performers.

Expert performers

As competent performers continue to gain experience performing tasks and troubleshooting problems, they increase the
richness of their understanding of how the product interface supports their ability to reach their goals. Competent
performers become experts when they are highly motivated, use the product frequently and as an integral part of their
jobs or personal activities and hobbies, have considerable subject matter knowledge, and are skilled at solving their own
problems and the problems of others. Figure 3-17 highlights the characteristics typically found with expert performers.

• Focus on developing a comprehetnsive and consistent mental model of the product functionality and the interface
• Ability to understand complex problems and find solutions
• Interest in learning about concepts and theories behind a product's design and use
• Interest in interacting with other expert users

Figure 3-17 Characteristics of expert performers.

In most instances you will find few experts in your user community. Most people simply lack the time or motivation to
become expert in more than one area and with more than a small number of products.
We can often identify expert performers by the number of competent performers who refer us to them. The experts are
the few individuals in an organization who are clearly recognized as more skilled in that particular area than anyone else.

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

They solve the most complex problems and often do the most difficult and infrequently performed tasks. They are the
people who work on the support lines, teach the workshops, and serve as consultants for the rest of the user community.
Expert performers have the kind of comprehensive understanding of the whole that allows them to create their own more
efficient ways to perform tasks. They are continually experimenting to increase their understanding. They often own five or
six books on the subject and read the journals and magazines that discuss their areas of interest and the products and
tools they use or need to know about. There may be little in task-oriented documentation to interest expert users, but they
are often skilled and frequent users of documentation. They use the information to gain the small insights that will enrich

their understanding and to learn to perform new tasks, especially with new versions of the product.

Expert performers are often engaged in a dialogue with product engineers and interface designers, pushing the envelope of the functions the product will perform and asking for alternative approaches to be included in the interface. Because they are already using the products to the fullest extent, they are always asking for more. The danger, of course, is that you may allow yourself, as a designer, to spend too much time catering to the needs of people at this stage of use.

Certainly your most expert customers will assist you in moving your product and its interface into new, uncharted areas. However, if they represent only a small percentage of your users, as is likely, they may sap your energies and move your attention from the needs of less proficient users. In fact, one of the dangers of enlisting expert users to serve on user requirements panels or to take part in focus groups to define new product features is that they will preempt your design time. For many designers and expert users, talking together is fun. Of all the user groups, experts may be the most like the designers. The fallacy of relying on the experts, even if you enjoy being with them most, is that they are not representative of most of your users. Figure 3-18 reminds you to pay attention to all four stages of use.

JoAnn worked on one interface design where the engineers spent most of their design time working on the exceptional and obscure tasks that some users wanted to perform. As a result, the primary functions served by the product and of interest to most users were neglected. The interface was difficult to understand and use, proving inaccessible to novices, advanced beginners, and even competent performers.

Applying the stages of use to your field studies

As you plan your field studies, consider how you will identify the stages of use within your user community. Plan your site visits (see chapters 7 and 8) so that you will see how novices, advanced beginners, competent performers, and experts are distributed among your users. If you have no users yet, consider finding out about the characteristics of potential users. Are they likely to be infrequent or frequent users? What subject-matter knowledge will they bring with them? What knowledge and experience in the technology will they bring with them? Do they have experience using similar products and interfaces? How much training and documentation will they require to move through the stages?

Figure 3-18 You will find users at all four stages of use, but many users will always remain advanced beginners and few users will ever become experts.

Référence du manuel : User and Task Analysis for Interface Design, Wiley, JoAnn T.Hackos & Janice C. Redish

Try to avoid having your site visits dominated by any one type of user. Some of your customers may want to steer you to their inhouse experts because competent performers and experts are likely to be more motivated and articulate in defining the problems with a product and its interface. Also, they want to show off their best performers. However, you should insist on seeing less competent but more typical users if advanced beginners are the most typical people among your

users.As you construct the profiles of your users, identify them in terms of stages of use. Consider what percentage of
your user community is likely to fall within each stage. Consider how your users are likely to move through the stages in
their interaction with your product. Consider the possible relationships between stages of use and other user
characteristics such as subject-matter knowledge, technology knowledge, and personal characteristics like learning style,
all of which were discussed in chapter 2.

As you can see, developing a picture of stages of use provides a link between the characteristics of your users, especially
their prior knowledge and experience, and your task analysis. If you observe only novices, you are unlikely to gain much
insight into the performance of more than a few tasks. If you study only experts, you will have difficulty designing for those
who are unfamiliar either with similar tools or with the subject matter. One word of caution. Do not assume that your own
personal characteristics are common to your users. You may prefer to read all the documentation before you begin to use
a new software application, but that does not mean that all your users will choose the same course. You may prefer
hacking your way through a new software application interface, but your users may be more timid about first-time use and
prefer to have some training or built-in job aid as a way to start.

References cited in the chapter

Benyon, David, The role of task analysis in systems design, *Interacting with* Computers, 4 (1), April 1992: 102-123.

Diaper, Dan and Addison, Mark, Task analysis and systems analysis for software development, Interacting with Computers, 4 (1),April 1992: 124-139.

Dray, Susan and Mrazek, Deborah, A day in the life of a family: An international ethnographic study, in Field *Methods Casebook* for *Software Design,* edited by Dennis W ixon and Judith Ramey, NY: John Wiley & Sons, 1996a, 145-156.

Dray, Susan and Mrazek, Deborah, A day in the life: Studying context across cultures, in *International User* Interfaces, edited by Elisa M. del Galdo and Jakob Nielsen, NY: John Wiley & Sons, 1996b, 242-256.

Dreyfus, Hubert L. and Dreyfus, Stuart E., *Minds over* Machines, NY: Macmillan, 1986.

Elm, Pelle, *Work-Oriented Design* of Computer Artifacts, Hillsdale, NJ: Lawrence Erlbaum Associates, 1989.

Elm, Pelle, Scandinavian design: On participation and skill, in Participatory *Design: Principles and Practices,* edited by Douglas Schuler and Aki Namioka, Hillsdale, NJ: Lawrence Erlbaum Associates, 1993, 41-77.

Juhl, Diane, Using field-oriented design techniques to develop consumer software products, in *Field Methods Casebook for Software Design,* edited by Dennis W ixon and Judith Ramey, NY: John Wiley & Sons, 1996, 215-228.